Towards joint decoding of binary Tardos fingerprinting codes

Peter Meerwald and Teddy Furon

Abstract

The class of joint decoder of probabilistic fingerprinting codes is of utmost importance in theoretical papers to establish the concept of fingerprint capacity [1]–[3]. However, no implementation supporting a large user base is known to date. This article presents an iterative decoder which is, as far as we are aware of, the first practical attempt towards joint decoding. The discriminative feature of the scores benefits on one hand from the side-information of previously accused users, and on the other hand, from recently introduced universal linear decoders for compound channels [4]. Neither the code construction nor the decoder make precise assumptions about the collusion (size or strategy). The extension to incorporate soft outputs from the watermarking layer is straightforward. An extensive experimental work benchmarks the very good performance and offers a clear comparison with previous state-of-the-art decoders.

Index Terms

Traitor tracing, Tardos codes, fingerprinting, compound channel.

I. Introduction

Traitor tracing or active fingerprinting has witnessed a flurry of research efforts since the invention of the now well-celebrated Tardos codes [5]. The codes of G. Tardos are optimal in the sense that the code length m necessary to fulfill the following requirements (n users, c colluders, probability of accusing at least one innocent below P_{fp}) has the minimum scaling in $\Omega(c^2 \log n P_{\mathsf{fp}}^{-1})$.

A first group of articles analyses such probabilistic fingerprinting codes from the viewpoint of information theory. They define the worst case attack a collusion of size c can produce, and also the best counter-attack. The main achievement is a saddle point theorem in the game between the colluders and the code designer which establishes the concept of fingerprinting capacity $C(c)$ [1]–[3]. Roughly speaking, for a maximum size of collusion c, the maximum number of users exponentially grows with m with an exponent equal to $C(c)$, to guarantee vanishing probabilities of error asymptotically as the code length increases. Sec. II summarizes these elements of information theory.

P. Meerwald and T. Furon are with INRIA Rennes, France; e-mail: {peter.meerwald, teddy.furon}@inria.fr.

EDICS Category: WAT-FING

Our point of view is much more practical and signal processing oriented. Thanks to an appropriate watermarking technique, m bits have been hidden in the distributed copies. At the time a pirated version is discovered, the content has been distributed to n users. Our goal is to identify some colluders under the strict requirement that the probability of accusing innocents is below P_{fp}. It is clear that we are not in an asymptotic setup since m and n are fixed. The encoder and the decoder are not informed of the collusion size and its attack, therefore there is no clue whether the actual rate $R = m^{-1}\log_2 n$ is indeed below capacity $C(c)$.

A second group of research works deals with decoding algorithms. Here, a first difficulty is to compute user scores that are as discriminative as possible. A second difficulty is to set a threshold such that one can reliably accuse users who are part of the collusion. These two steps are not easy since the decoder does not know the size and the attack of the collusion. Sec. III sums up the past approaches which are mainly based on single decoders. It also motivates our decoder based on compound channel theory and the use of a rare event estimator.

A third difficulty is to have a fast implementation of the accusation algorithm in order to face a large-scale set of users. A main advantage of some fingerprinting schemes based on error-correcting codes is to offer an accusation procedure with runtime polynomial in m [6], [7]. In comparison, the well-known Tardos-Škorić single decoder is an exhaustive search of complexity $O(nm)$ [5], [8]. Since in theory n can asymptotically be in the order of 2^{mR}, decoding of Tardos codes might be intractable. Again, we do not consider such a theoretical setup, but we pay attention to maintain an affordable decoding complexity for orders of magnitude met in practical applications.

Sec. IV focuses on the iterative architecture of our joint decoder based on three primitives: channel inference, score computation, and thresholding. Its iterative nature stems from two key ideas: i) the codeword of a newly accused user is integrated as a side information for the next iterations, ii) joint decoding is manageable on a short list of suspects. Sec. V provides an extension to soft decoding. In Sec. VI we present our experimental investigations with a comparison with related works for typical values of (m, n). This shows the benefit of our decoder: better decoding performance with acceptable runtime in practical scenarios.

II. Tardos code and the collusion model

We briefly review the construction and some known facts about Tardos codes.

A. Construction

The binary code is composed of n codewords of m bits. The codeword $\mathbf{x}_j = (x_j(1), \cdots, x_j(m))^T$ identifying user $j \in \mathcal{U} = [n]$, where $[n] := \{1, \dots, n\}$, is composed of m binary symbols independently drawn at the code construction s.t. $\mathbb{P}(x_j(i) = 1) = p_i$, $\forall i \in [m]$. At initialization, the auxiliary variables $\{p_i\}_{i=1}^m$ are independent and identically drawn according to distribution $f(p) : [0,1] \to \mathbb{R}^+$. Both the code $\Xi = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ and the auxiliary sequence $\mathbf{p} = (p_1, \dots, p_m)^T$ must be kept as secret parameters.

B. Collusion attack

The collusion attack or collusion channel describes the way the c colluders $\mathcal{C} = \{j_1, \dots, j_c\}$ merge their binary codewords $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_c}$ to forge the binary pirated sequence $\mathbf{y}$. It is usually modelled as a memoryless discrete multiple access channel, which is fair in the sense that all colluders participate equally in the forgery. This assumption comes from the fact that the worst case attacks are indeed memoryless for Tardos codes where symbols are generated independently, [9, Lemma 3.3]. Moreover, in a detect-many scenario, there is no hope in identifying almost idle colluders if the attack is not fair [9, Lemma 3.2].

This leads to a $2 \times (c+1)$ probability transition matrix $[\mathbb{P}(Y|\varPhi)]$ where $\varPhi = \sum_{j \in \mathcal{C}} X_j$ is a random variable counting the number of '1' the colluders received out of c symbols. A common parameter of the collusion attack on binary codes is denoted by the vector $\boldsymbol{\theta}_c = (\theta_c(0), \ldots, \theta_c(c))^T$ with $\theta_c(\varphi) = \mathbb{P}(Y = 1|\varPhi = \varphi)$. The usual working assumption, so-called *marking assumption* [10], imposes that $\theta_c(0) = 1 - \theta_c(c) = 0$. The set of collusion attacks that c colluders can lead under the marking assumption is denoted by Θ_c:

$$\Theta_c = \{\boldsymbol{\theta} \in [0,1]^{c+1}, \theta(0) = 1 - \theta(c) = 0\}. \tag{1}$$

Examples of attacks following this model are given, for instance, in [11].

C. Accusation

Denote $\mathcal{A} \subset \mathcal{U}$ the set of users accused by the decoder. The probability of false positive is defined by $P_{\mathsf{fp}} = \mathbb{P}(\mathcal{A} \not\subset \mathcal{C})$. In practice, a major requirement is to control this feature so that it is lower than a given significance level.

In a detect-one scenario, $\mathcal{A}$ is either a singleton, or the empty set. A good decoder has a low probability of false negative defined by $P_{\mathsf{fn}} = \mathbb{P}(\mathcal{A} = \emptyset)$. In a detect-many scenario, several users are accused, and a possible figure of merit is the number of caught colluders: $|\mathcal{A} \cap \mathcal{C}|$. In the literature, there exists a third scenario, so-called detect-all, where a false negative happens if at least one colluder is missed. This article only considers the first two scenarios.

D. Guidelines from information theory

This article does not pretend to any new theoretical contribution, but presents some recent elements to stress guidelines when designing our practical decoder.

A *single decoder* computes a score per user. It accuses users whose score is above a threshold (detect-many scenario) or the user with the biggest score above the threshold (detect-one scenario). Under both scenarios and provided that the collusion is fair, the performance of such decoders is theoretically bounded by the achievable rate $R_S(f, \boldsymbol{\theta}_c) = I(X; Y|P, \boldsymbol{\theta}_c) = \mathbb{E}_{P \sim f}[I(X; Y|p, \boldsymbol{\theta}_c)]$ [9, Th. 4.1]. A fundamental result is that, for a given collusion size c, there exists an equilibrium $(\breve{f}_{c,S}, \breve{\boldsymbol{\theta}}_{c,S})$ to the max-min game between the colluders (who select $\boldsymbol{\theta}$) and the code designer (who selects f) as defined by $\max_f \min_{\boldsymbol{\theta} \in \Theta_c} R_S(f, \boldsymbol{\theta})$ in [1, Th. 4].

A *joint decoder* computes a score per subset of $\ell \leq c$ users and accuses the users belonging to subsets whose score is above a threshold or only the most likely guilty amongst these users. Under both scenarios and provided that the collusion is fair, the performance of such decoders is theoretically bounded by the achievable rate $R_J(f, \boldsymbol{\theta}_c) = \ell^{-1} I(\varPhi; Y|P, \boldsymbol{\theta}_c) = \ell^{-1} \mathbb{E}_{P \sim f}[I(\varPhi; Y|p, \boldsymbol{\theta}_c)]$ [9, Th. 3.3]. $\varPhi$ denotes the random variable sum of the subset user symbols. Moreover, for a given collusion size c, there also exists an equilibrium $(\breve{f}_{c,J}, \breve{\boldsymbol{\theta}}_{c,J})$ to the max-min game $\max_f \min_{\boldsymbol{\theta}\in\Theta_c} R_J(f,\boldsymbol{\theta})$ [1, Th. 4].

Asymptotically, as $c \to +\infty$, both $\breve{f}_{c,J}$ and $\breve{f}_{c,S}$ converge to $f_T(p) = 1/(\pi\sqrt{p(1-p)})$, the distribution originally proposed by G. Tardos [1, Cor. 7], and both $\min_{\boldsymbol{\theta}} R_J(f_T,\boldsymbol{\theta})$ and $\min_{\boldsymbol{\theta}} R_S(f_T,\boldsymbol{\theta})$ quickly approach the equilibrium value of the respective max-min game [1, Fig. 2]. Yet, the code designer needs to bet on a collusion size c' in order to use the optimal distribution $\breve{f}_{c',S}$ (or $\breve{f}_{c',J}$ if the decoder is joint). Integer c' plays the role of a desired security level.

Despite the division by ℓ in the expression of $R_J(f,\boldsymbol{\theta})$, it appears that $R_S(f,\boldsymbol{\theta}) \leq R_J(f,\boldsymbol{\theta})$, $\forall\boldsymbol{\theta}$ [9, Eq. (3.4)]. This tells us that a joint decoder is theoretically more powerful than a single decoder. However, a joint decoder needs to compute $O(n^\ell)$ scores since there are $\binom{n}{\ell}$ subsets of size ℓ. This complexity is absolutely intractable for large-scale applications even for a small ℓ. This explains why, so far, joint decoders were only considered theoretically to derive fingerprinting capacity. Our idea is that there is no need to consider all these subsets since a vast majority is only composed of innocent users. Our decoder iteratively prunes out users deemed as innocents and considers the subsets over the small set of remaining suspects.

This iterative strategy results in a decoder which is a mix of single and joint decoding. Unfortunately, it prevents us from taking advantage of the game theory theorems mentioned above. We cannot find the optimal distribution f and the worst collusion attack against our decoder. Nevertheless, our decoder works with any distribution f under some conditions stated in Sec. III. For all these reasons, the experiments of Sec. VI are done with the most common Tardos distribution f_T.

M. Fernandez and M. Soriano proposed an iterative accusation process of an error correcting code based fingerprinting scheme [7]. Each iteration takes advantage of the codewords of colluders already identified in the previous iterations. The same idea is possible with Tardos probabilistic fingerprinting code. This is justified by the fact that the side information Δ, defined as the random variable sum of the already identified colluder symbols, increases the mutual information: $I(\Phi;Y|P,\boldsymbol{\theta}_c) \leq I(\Phi;Y|P,\boldsymbol{\theta}_c,\Delta)$. Indeed, side information helps more than joint decoding as proved by [9, Eq. (3.3)].

The above guidelines can be summarized as follows: use the continuous Tardos distribution f_T for code construction, integrate the codewords of accused users as side information and finally use a joint decoder on a short list of suspects.

III. A SINGLE DECODER BASED ON COMPOUND CHANNEL THEORY AND RARE EVENT ANALYSIS

This section first reviews some single decoders and presents new decoders based on compound channel theory and rare event analysis. The first difficulty is to compute a score per user such that the colluders are statistically well separated from the innocents scores. The second difficulty is to set a practical threshold such that the probability of false positive is under control.

Detection theory tells us that the score given by the Log-Likelihood Ratio (LLR):

$$s_j = \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i), \boldsymbol{\theta}_c)}{\mathbb{P}(y(i)|\boldsymbol{\theta}_c)}, \tag{2}$$

is optimally discriminative in the Neyman-Pearson sense to decide the guiltiness of user j. Yet, the LLR needs the knowledge of the true collusion attack $\boldsymbol{\theta}_c$ which prevents the use of this optimal single decoder in practical settings. Some papers proposed a so-called 'Learn and Match' strategy using the LLR score tuned on an estimation $\hat{\boldsymbol{\theta}}$ of the attack channel [11]. Unfortunately, a lack of identifiability obstructs a direct estimation from $(\mathbf{y}, \mathbf{p})$ (see Sec. III-B). Indeed, the estimation is sound only if c is known, and if the number of different values taken by p is bigger[1] or equal than $c-1$: $\mathbb{P}(Y = 1|\boldsymbol{\theta}, p)$ is a polynomial in p of degree at most c (see (14) with $u = 0$ and $v = 0$) going from point $(0, 0)$ to $(1, 1)$, we need $c - 1$ more points to uniquely identify this polynomial. To overcome this lack of information about c, an Expectation-Maximization (E.-M.) approach has been proposed but it is not satisfactory since it does not scale well with the number of users [11]. Moreover, the setting of the threshold was not addressed.

On the other hand, there are decoders that do not adapt their score computation to the collusion. This is the case of the score computation originally proposed by G. Tardos [5], and later-on improved by B. Škorić *et al.* [8]. It has an invariance property: its statistics, up to the second order, do not depend on the collusion attack channel $\boldsymbol{\theta}$, but only on the collusion size c [12]. Thanks to this invariance, whatever the collusion attack is, there exists a threshold τ guaranteeing a probability of false positive below P_{fp} while keeping the false negative away from 1 provided that the code is long enough, *i.e.* $m = \Omega(c^2 \log n P_{\mathsf{fp}}^{-1})$. However, there is a price to pay: the scores are clearly less discriminative than the LLR.

Some theoretical papers [13, Sec. V] [9, Sec. 5.2] promote another criterion, so-called 'universality', for the design of decoders. The performance (usually evaluated as the achievable rate or the error exponent) when facing a collusion channel $\boldsymbol{\theta}_c$ should not be lower than the performance against the worst attack $\boldsymbol{\theta}_c^\star$. In a sense, it is a clear warning to the 'Learn and Match' strategy. Suppose that $\boldsymbol{\theta}_c \neq \boldsymbol{\theta}_c^\star$ and that, for some reasons, the estimation of the collusion attack is of poor quality. In any case, a mismatch between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}_c$ should not ruin the performance of the decoder to the point it is even lower than what is achievable under the worst attack $\boldsymbol{\theta}_c^\star$. The above cited references [9], [13] recommend the single universal decoder based on the empirical mutual information $I(\mathbf{x}; \mathbf{y}|\mathbf{p})$ (or empirical equivocation for joint decoder). The setting of the threshold depends on the desired error exponent of the false positive rate. Therefore, it is valid only asymptotically.

To summarize, there have been two approaches: adaptation or non-adaptation to the collusion process. The first class is not very well grounded since the estimation of the collusion is an issue and the impact of a mismatch has to be studied. The second approach is more reliable, but with a loss of discrimination power compared to the optimal LLR. The next sections presents two new decoders belonging to both approaches based on the compound channel theory.

[1] This is the case in this article since we opt for the continuous Tardos distribution f_T.

A. *Some elements on compound channels*

Recently, in the setup of digital communication through compound channels, E. Abbe and L. Zheng [4] proposed universal decoders which are linear, *i.e.* in essence very simple. This section summarizes this theory and the next one proposes two applications for Tardos single decoders.

A compound channel is a set $\mathcal{S}$ of channels, say discrete memoryless channels $X \in \mathcal{X} \rightarrow Y \in \mathcal{Y}$ defined by their probability transition matrix $W_\theta = [\mathbb{P}(Y|X,\theta)]$ parameterized by $\theta \in \Theta$. The coder shares a code book $\Xi = \{\mathbf{x}_j\}_{j=1}^n \in \mathcal{X}^{m\times n}$ with the decoder. Its construction is assumed to be a random code realization from a provably good mass distribution P_X. After receiving a channel output $\mathbf{y} \in \mathcal{Y}^m$, a decoder computes a score per codeword $\mathbf{x}_j$, $j \in [n]$, and yields the message associated with the codeword with the biggest score. The decoder is linear if the score has the following structure:

$$s_j = \sum_{i=1}^{m} d(x_j(i), y(i)), \tag{3}$$

with $d(\cdot,\cdot) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. For instance, score (2), so-called MAP decoder in digital communications [4], is linear with $d(x,y) = \log(\mathbb{P}(y|x,\theta)/\mathbb{P}(y|\theta))$. However, in the compound channel setup, the decoder does not know through which channel of $\mathcal{S}$ the codeword has been transmitted, and therefore it cannot rely on the MAP.

We are especially interested in two results. First, if $\mathcal{S}$ is *one-sided* (see Def. 1 below), then the MAP decoder tuned on the worst channel $W_{\theta^\star}$ is a linear universal decoder [4, Lemma 5]. If $\mathcal{S} = \bigcup_{k=1}^{K} \mathcal{S}_k$ with K finite and $\mathcal{S}_k$ one-sided $\forall k \in [K]$, then the following *generalized* linear decoder is universal [4, Th. 1] and the score of a codeword is the maximum of the K MAP scores tuned on the worst channel $W_{\theta_k^\star}$ of each $\mathcal{S}_k$:

$$s_j = \max_{k\in[K]} \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i), \theta_k^\star)}{\mathbb{P}(y(i)|\theta_k^\star)}. \tag{4}$$

Definition 1 (One-sided set, Def. 3 of [4]): A set $\mathcal{S}$ is one-sided with respect to an input distribution P_X

- if the following minimizer is unique:

$$W_{\theta^\star} = \arg \min_{\theta\in \mathsf{cl}(\Theta)} \mathcal{I}(P_X, \theta), \tag{5}$$

  with $\mathcal{I}(P_X,\theta)$ the mutual information $I(X;Y)$ with $(X,Y) \sim P_X \circ W_\theta$ (where $P \circ W$ denotes the joint distribution with P the distribution of X and W the conditional distribution), and $\mathsf{cl}(\Theta)$ the closure of Θ,

- and if, $\forall \theta \in \Theta$,

$$\begin{aligned} D(P_X \circ W_\theta || P_X \times P_{Y,\theta^\star}) \geq & D(P_X \circ W_\theta || P_X \circ W_{\theta^\star}) + \\ & D(P_X \circ W_{\theta^\star} || P_X \times P_{Y,\theta^\star}). \end{aligned} \tag{6}$$

  with $D(\cdot||\cdot)$ the Kullback-Leibler distance, $P_{Y,\theta}$ the marginal of Y induced by $P_X \circ W_\theta$, and $P_X \times P_{Y,\theta}$ the product of the marginals.

B. *Application to single Tardos decoders*

Contrary to the code construction phase, it is less critical at the decoding side to presume that the real collusion size c is less or equal to a given parameter c_{max}. This parameter can be set to the largest number of colluders the fingerprinting code can handle with a reasonable error probability knowing (m, n). Another argument is that this assumption is not definitive. If the decoding fails because the assumption does not hold true, nothing prevents us to re-launch decoding with a bigger c_{max}. Let us assume $c \leq c_{\mathsf{max}}$ in the sequel.

A first application of the work [4] is straightforward: The collusion channel belongs to the set $\bigcup_{k=2}^{c_{\mathsf{max}}} \Theta_k$ as defined (1), and thanks to [4, Lemma 4] each convex set Θ_k is one-sided. According to [4, Th. 1], the decoder based on the following score is universal:

$$s_j = \max_{k \in [2,\ldots,c_{\mathsf{max}}]} \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i), \boldsymbol{\theta}^{\star}_{k,f_T})}{\mathbb{P}(y(i)|\boldsymbol{\theta}^{\star}_{k,f_T})}, \tag{7}$$

where $\boldsymbol{\theta}^{\star}_{k,f_T} = \arg\min_{\Theta_k} R_S(f_T, \boldsymbol{\theta})$, $\forall k \in [2, \ldots, c_{\mathsf{max}}]$. This decoder does not adapt its score computation to the collusion attack.

The second application is more involved as the lack of identifiability turns to our advantage. The true collusion channel $\boldsymbol{\theta}_c$ has generated data $\mathbf{y}$ distributed as $\mathbb{P}(y|p, \boldsymbol{\theta}_c)$. Let us define the class $\mathcal{E}(\boldsymbol{\theta}_c) = \{\tilde{\boldsymbol{\theta}} | \mathbb{P}(y|p, \tilde{\boldsymbol{\theta}}) = \mathbb{P}(y|p, \boldsymbol{\theta}_c), \forall (y,p) \in \{0,1\} \times [0,1]\}$. Thanks to [14, Prop. 3], we know that $\mathcal{E}(\boldsymbol{\theta}_c)$ is not restricted to the singleton $\{\boldsymbol{\theta}_c\}$ since for any $c' > c$ there exists one $\tilde{\boldsymbol{\theta}}_{c'} \in \mathcal{E}(\boldsymbol{\theta}_c)$. This holds especially for c_{max}. Asymptotically with the code length, the consistent Maximum Likelihood Estimator (MLE) parameterized on c_{max}, as defined in (16), yields an estimation $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} \approx \tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}} \in \mathcal{E}(\boldsymbol{\theta}_c)$ with increasing accuracy. This estimation is not reliable because $c \neq c_{\mathsf{max}}$ a priori. Therefore, we prefer to refer to $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$ as a collusion inference rather than a collusion estimation, and the scoring uses this inference as follows:

$$s_j = \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i), \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}{\mathbb{P}(y(i)|\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}. \tag{8}$$

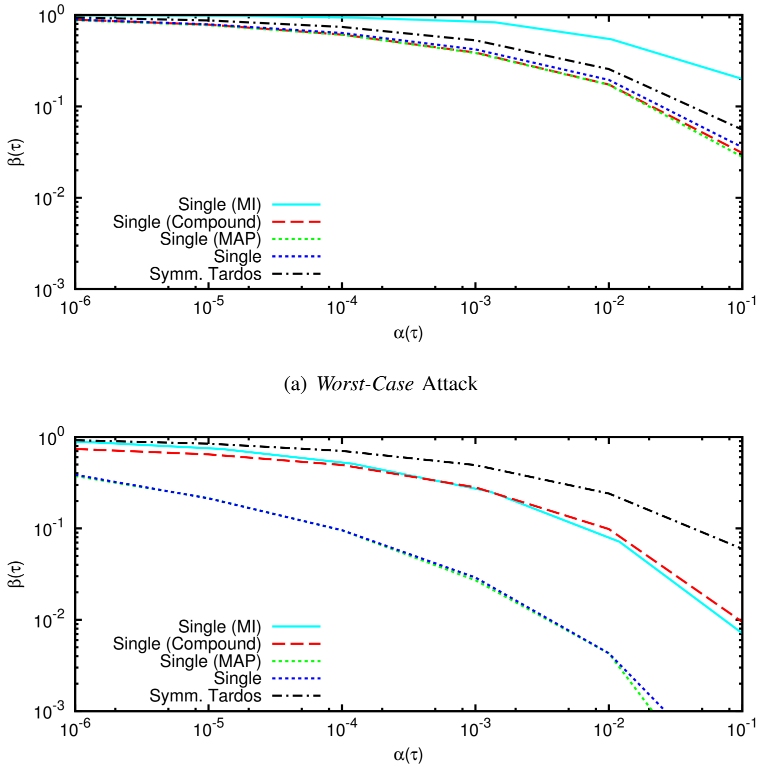
Suppose that the MLE tuned on c_{max} provides a perfect inference $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} = \tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$, we then succeed to restrict the compound channel to the discrete set $\mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c)$ which we define as the restriction of $\mathcal{E}(\boldsymbol{\theta}_c)$ to collusions of size $\tilde{c} \leq c_{\mathsf{max}}$. Appendix A shows that $\mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c)$ is one-sided, and its worst attack is indeed $\tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$. Lemma 5 of [4] justifies the use of the MAP decoder (2) tuned on $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$. Its application leads to a more efficient decoder since $R_S(f_T, \tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}) \geq R_S(f_T, \boldsymbol{\theta}^{\star}_{c_{\mathsf{max}},f_T})$. This decoder pertains to the approach based on score adaptation, with the noticeable advantages: it is better theoretically grounded and it is far less complex than the iterative E.-M. decoder of [11].

Figure 1 illustrates the Receiver Operating Characteristics (ROC) per user for the single decoders discussed so far with $m = 512$ and $c = 5$ colluders performing *worst-case* (*i.e.* minimizing $R_S(f_T, \boldsymbol{\theta})$ over Θ_5) and *majority* attack ($\theta_{5,\mathsf{maj}} = (0,0,0,1,1,1)^T$). For this figure, the false positive $\alpha(\tau)$ and the false negative $\beta(\tau)$ are defined *per user* as follows:

$$\alpha(\tau) = \mathbb{P}(s(\mathbf{x}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p}) > \tau), \tag{9}$$

$$\beta(\tau) = \mathbb{P}(s(\mathbf{x}_{j_1}, \mathbf{y}, \mathbf{p}) \leq \tau), \tag{10}$$

where $\mathbf{x}_{\mathsf{inn}}$ is a random variable denoting the codeword of an innocent user and $\mathbf{x}_{j_1}$, the codeword of the first colluder. The *single* decoder is tuned on the collusion inference $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$ (with $c_{\mathsf{max}} = 8$) and performs almost as



(a) *Worst-Case* Attack

(b) *Majority* Attack

Fig. 1. ROC plot for several decoders; $m = 512$, $c = 5$, $c_{\mathsf{max}} = 8$. Single (MI) is the decoder based on empirical mutual information [9], Single (Compound) relates to (7), Single (MAP) is (2), Single is the LLR on $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$ (8), and Symm. Tardos is the symmetric version of the G. Tardos scores proposed by B. Škorić *et al.* in [8].

good as the MAP decoder having knowledge of $\boldsymbol{\theta}$. The ROC of the symmetric Tardos score is invariant w.r.t. the collusion attack. The generalized linear decoder of (7) denoted *compound* takes little advantage of the fact that the majority attack is much milder than the worst attack. For a fair comparison, the single decoder based on the empirical mutual information [9] assumes a Tardos distribution uniformly quantized to 10 bins; better results (yet still below the *single* decoder) can be obtained when tuned to the optimal discrete distribution for $c = 5$ colluders [15].

The similarities between compound channel and fingerprinting has been our main inspiration, however some differences prevent any claim of optimality. First, in the compound channel problem, there is a unique codeword that has been transmitted, whereas in fingerprinting, $\mathbf{y}$ is forged from c codewords like in a multiple access channel. Therefore, the derived single decoders are provably good for chasing a given colluder (detect-one scenario), but they might not be the best when looking for more colluders (detect-many scenario). The second difference is that the decoder should give up when not confident enough rather than taking the risk of being wrong in accusing an innocent. The setting of a threshold is clearly missing for the moment.

C. Rare event analysis

This section explains how we set a threshold τ in accordance with the required P_{fp} thanks to a rare event analysis. Our approach is very different than [13] [9] [2] [5] where a theoretical development either finds a general threshold suitable when facing a collusion of size c, or equivalently, where it claims a reliable decision when the rate is below the capacity which depends on c. Our threshold does not need the value of c but it only holds for a given couple $(\mathbf{p}, \mathbf{y})$ and a known n. Once these are fixed, the scoring $s_j = s(\mathbf{x}_j, \mathbf{y}, \mathbf{p})$ is a deterministic function from $\{0,1\}^m$ to $\mathbb{R}$. Since the codewords of the innocent users are i.i.d. and $c \ll n$, we have:

$$\begin{aligned} P_{\mathsf{fp}} &= 1 - (1 - \mathbb{P}(s(\mathbf{x}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p}) > \tau))^{n-c} \\ &\approx n \cdot \mathbb{P}(s(\mathbf{x}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p}) > \tau). \end{aligned} \tag{11}$$

The number of possible codewords can be evaluated as the number of typical sequences, *i.e.* in the order of $2^{m\mathbb{E}_{P\sim f}[h_b(p)]}$, with $h_b(p)$ the entropy in bits of a Bernoulli random variable $B(p)$. $\mathbb{E}_{P\sim f_T}[h_b(p)] \approx 0.557$ bits, which leads to a far bigger number of typical sequences than n (say $m \geq 300$ and $n \leq 10^8$ in practice). This shows that plenty of codewords have not been created when a pirate copy is found. Therefore, we consider them as occurrences of $\mathbf{x}_{\mathsf{inn}}$ since we are sure that they have not participated in the forgery of $\mathbf{y}$. The idea is then to estimate τ s.t. $\mathbb{P}(s(\mathbf{x}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p}) > \tau) = n^{-1} P_{\mathsf{fp}}$ thanks to a Monte Carlo simulation with newly created codewords.

The difficulty lies in the order of magnitude. Some typical requirements are $n \approx 10^6$ and $P_{\mathsf{fp}} = 10^{-4}$, hence the estimation of τ corresponding to a probability as small as 10^{-10}. This is not tractable with a basic Monte Carlo on a regular computer. However, the new estimator based on rare event analysis proposed in [16] performs remarkably fast within this range of magnitude. It produces $\hat{\tau}$ and a C-% confidence interval[2] $[\tau^-, \tau^+]$. In our decoder, we compare the scores to τ^+ (*i.e.* a pessimistic estimate of τ) to ensure a total false positive probability lower than P_{fp}. Last but not least, this approach works for any single decoder.

IV. Iterative, Joint decoding algorithm

This section extends the single decoder based on the collusion inference $\boldsymbol{\theta}_{c_{\mathsf{max}}}$ towards joint decoding, thanks to the guidelines of Sec. II-D. Preliminary results about these key ideas were first presented in [17] and [18]. A schematic overview of the iterative, joint decoder is shown in Fig. 2.

A. Architecture

The first principle is to iterate the score computation and include users accused in previous iterations as side-information to build a more discriminative test. Let $\mathcal{U}_{\mathsf{SI}} = \emptyset$ denote the initially empty set of accused users. In each iteration we aim at identifying a (possibly empty) set of users $\mathcal{A} = \{j \in \mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}} | s_j > \tau\}$ and then update $\mathcal{U}_{\mathsf{SI}}$ with $\mathcal{A}$.

Second, we additionally compute scores for subsets of t users of $\mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}$, $t \leq c_{\mathsf{max}}$. Obviously, there are $\binom{|\mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}|}{t}$ such subsets. As n is large, enumerating and computing a score for each subset is intractable even for small t. The

[2] We are C-% sure that the true τ lies in this interval.

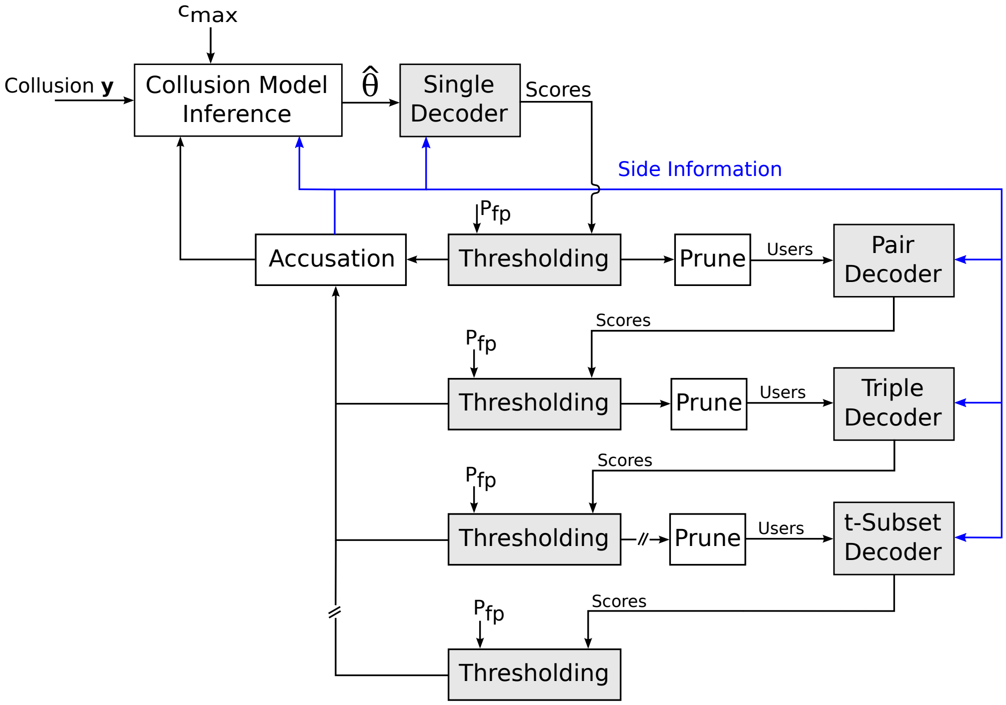


Fig. 2. Overview of the iterative, side-informed joint Tardos fingerprint decoder.

idea here is to find a restricted set $\mathcal{U}^{(t)} \subseteq \mathcal{U}\backslash\mathcal{U}_{\mathsf{SI}}$ of $n^{(t)} = |\mathcal{U}^{(t)}|$ users that are the most likely to be guilty and keep $p^{(t)} = \binom{n^{(t)}}{t}$ approximately constant and within our computation resources. We gradually reduce $n^{(t)}$ by pruning out users who are unlikely to be colluder when going from single ($t = 1$) decoding, to pair ($t = 2$) decoding, etc. If $n^{(t)} = O(n^{1/t})$, then score computation of t-subsets over the restricted user set is within $O(n)$ just like for the single decoder.

Initially, the joint pair-decoder starts with the list of users ranked by the scores derived from the single decoder in decreasing order, *i.e.* the top-ranked user is most likely to be a colluder. Later on, the joint t-subset decoder produces a new list of scores computed from subsets of t users which – according to theoretical results [2], [9] – are more discriminative as t increases. Denote $\mathcal{T}^{\diamond} \subseteq \mathcal{U}^{(t)}$ the t-subset of users with the highest score. Our algorithm tries to accuse the most likely colluder within $\mathcal{T}^{\diamond}$, and, if successful, updates $\mathcal{U}_{\mathsf{SI}}$ and continues with the single decoder. If no accusation can be made, the algorithm generates a new list of suspects $\mathcal{U}^{(t+1)}$ based on the ranking of joint scores that is fed to the subsequent $t + 1$ joint decoding stage.

In the detect-one scenario, iteration stops after the first accusation. We restrict the subset size to $t \leq t_{\mathsf{max}}$, with $t_{\mathsf{max}} = 5$. This is not a severe limitation as for moderately large c, the decoding performance advantage of the joint decoder quickly vanishes [9]. In the detect-many scenario, iteration stops when $|\mathcal{U}_{\mathsf{SI}}| \geq c_{\mathsf{max}}$ or t reaches $\min(t_{\mathsf{max}}, c_{\mathsf{max}} - |\mathcal{U}_{\mathsf{SI}}|)$ and no further accusation can be made. The set $\mathcal{U}_{\mathsf{SI}}$ then contains the user indices to be accused. Alg. 1 illustrates the architecture of the accusation process for the catch-many scenario.

The next sections describe the score computation, the accusation of a user and the inference of the collusion process in more details.

Algorithm 1 Iterative Joint Tardos Decoder.

```
Require: $\mathbf{y}$, $\Xi$, $\mathbf{p}$, $c_{\mathsf{max}}$, $t_{\mathsf{max}} \leq c_{\mathsf{max}}$, $n^{(t)}$, $P_{\mathsf{fp}}$
$\mathcal{U} \leftarrow \{j | 1 \leq j \leq n\}$, $\mathcal{U}_{\mathsf{SI}} \leftarrow \emptyset$
repeat
    $t \leftarrow 1$
    $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} \leftarrow \mathtt{infere}(\mathbf{y}, \mathbf{p}, \mathcal{U}_{\mathsf{SI}}, c_{\mathsf{max}})$
    $\mathbf{W} \leftarrow \mathtt{weights}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}, \mathcal{U}_{\mathsf{SI}})$
    $\mathbf{s} \leftarrow \mathtt{scores}(\mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}, \Xi, \mathbf{W})$
    $\tau \leftarrow \mathtt{threshold}(\mathbf{p}, \mathbf{W}, n^{-1} P_{\mathsf{fp}})$
    $\mathcal{A} \leftarrow \{j \in \mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}} | s_j > \tau\}$
    while $\mathcal{A} = \emptyset$ and $t < t_{\mathsf{max}}$ do
        $t \leftarrow t + 1$
        $\mathcal{U}^{(t)} \leftarrow \{j \in \mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}} | s_j > \mathtt{top}(\mathbf{s}, n^{(t)})\}$
        $\mathbf{W} \leftarrow \mathtt{weights}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}, \mathcal{U}_{\mathsf{SI}})$
        $\mathbf{s} \leftarrow \mathtt{scores}(\binom{\mathcal{U}^{(t)}}{t}, \Xi, \mathbf{W})$
        $\tau \leftarrow \mathtt{threshold}(\mathbf{p}, \mathbf{W}, \binom{n}{t}^{-1} P_{\mathsf{fp}}, t)$
        $\mathcal{T}^{\diamond} \leftarrow \arg\max_{\mathcal{T} \in \mathcal{U}^{(t)}} s_{\mathcal{T}}$
        if $s_{\mathcal{T}^{\diamond}} > \tau$ then
            for all $j \in \mathcal{T}^{\diamond}$ and while $\mathcal{A} = \emptyset$ do
                $\mathbf{W} \leftarrow \mathtt{weights}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}, \mathcal{U}_{\mathsf{SI}} \cup \{\mathcal{T}^{\diamond} \setminus j\})$
                $\tau' \leftarrow \mathtt{threshold}(\mathbf{p}, \mathbf{W}, n^{-1} P_{\mathsf{fp}})$
                $\mathcal{A} \leftarrow \{j | \mathtt{score}(j, \Xi, \mathbf{W}) > \tau'\}$
            end for
        end if
    end while
    $\mathcal{U}_{\mathsf{SI}} \leftarrow \mathcal{U}_{\mathsf{SI}} \cup \mathcal{A}$
until $\mathcal{A} = \emptyset$ or $|\mathcal{U}_{\mathsf{SI}}| \geq c_{\mathsf{max}}$
return $\mathcal{U}_{\mathsf{SI}}$
```

B. Score computation

For a t-subset $\mathcal{T}$, the accusation is formulated as a hypothesis test based on the observations $(\mathbf{y}, \{\mathbf{x}_j\}_{j \in \mathcal{T}})$ to decide between $\mathcal{H}_0$ (all $j \in \mathcal{T}$ are innocent) and $\mathcal{H}_1$ (all $j \in \mathcal{T}$ are guilty). The score is just the LLR tuned on the inference $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$ of the collusion process.

All these sequences are composed of independent random variables thanks to the code construction and the memoryless nature of the collusion. Moreover, the collusion only depends on the number of symbol '1' present in the codewords of a subset. Therefore, denote by $\boldsymbol{\delta}$ and $\boldsymbol{\varphi}$ the accumulated codewords corresponding to $\mathcal{U}_{\mathsf{SI}}$ and $\mathcal{T}$: $\boldsymbol{\delta} = \sum_{j\in\mathcal{U}_{\mathsf{SI}}} \mathbf{x}_j$ and $\boldsymbol{\varphi} = \sum_{j\in\mathcal{T}} \mathbf{x}_j$. We have $\forall i \in [m]$, $0 \leq \delta(i) \leq n_{\mathsf{SI}}$ and $0 \leq \varphi(i) \leq t$. Thanks to the linear structure of the decoder, the score for a subset $\mathcal{T}$ of t users is simply

$$s_{\mathcal{T}} = \sum_{i=1}^{m} W(\varphi(i), i), \tag{12}$$

where the $(t+1)\times m$ weight matrix $\mathbf{W}$ is pre-computed from $(\mathbf{y},\mathbf{p})$ taking into account the side information $\mathcal{U}_{\mathsf{SI}}$ so that $\forall(\varphi, i) \in \{0,\dots,t\}\times\{1,\dots,m\}$:

$$W(\varphi, i) = \log \frac{\mathbb{P}(y(i)|(\varphi, t), (\delta(i), n_{\mathsf{SI}}), p(i), \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}{\mathbb{P}(y(i)|(\delta(i), n_{\mathsf{SI}}), p(i), \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}. \tag{13}$$

For indices s.t. $y(i) = 1$, both the numerator and the denominator share a generic formula, $P(\varphi(i)+\delta(i), t+n_{\mathsf{SI}}, p(i), \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})$ and $P(\delta(i), n_{\mathsf{SI}}, p(i), \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})$ respectively, with

$$\begin{aligned} P(u, v, p, \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}) = & \sum_{k=u}^{c_{\mathsf{max}}-v+u} \hat{\theta}_{c_{\mathsf{max}}}(k)\cdot \\ & \binom{c_{\mathsf{max}}-v}{k-u} p^{k-u}(1-p)^{c_{\mathsf{max}}-v-k+u}. \end{aligned} \tag{14}$$

In words, this expression gives the probability that $y = 1$ knowing that the symbol '1' has been distributed to users with probability p, the collusion model $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$, and the identity of v colluders who have u symbols '1' and $v-u$ symbols '0'. For indices s.t. $y(i) = 0$ in (13), the numerator and the denominator need to be 'mirrored': $(P \to 1-P)$.

At iterations based on the single decoder: $t = 1$ and $\boldsymbol{\varphi} = \mathbf{x}_j$ for user j. If nobody has been deemed guilty so far, then $\delta(i) = n_{\mathsf{SI}} = 0, \forall i \in [m]$. This score is defined if $t + n_{\mathsf{SI}} \leq c_{\mathsf{max}}$. Therefore, for a given size of side-information, we cannot conceive a score for subsets of size bigger than $c_{\mathsf{max}} - n_{\mathsf{SI}}$. This implies that in the detect-many scenario, the maximal number of iterations depends on how fast $\mathcal{U}_{\mathsf{SI}}$ grows.

C. *Ranking users within a subset and joint accusation*

Let $\mathcal{T}^{\diamond}$ denote the t-subset with the highest score. We accuse one user in $\mathcal{T}^{\diamond}$ only if $s_{\mathcal{T}^{\diamond}} > \tau$. Let $\mathcal{T}_{\mathsf{inn}}$ denote a subset composed of innocent users. Using rare event analysis, τ is estimated s.t. $\mathbb{P}(s(\{\mathbf{x}_j\}_{j\in\mathcal{T}_{\mathsf{inn}}}, \mathbf{y}, \mathbf{p}) > \tau) = \binom{n}{t}^{-1} P_{\mathsf{fp}}$. This thresholding operation ensures that $\mathcal{T}^{\diamond}$ contains at least one colluder with a very high probability.

In order to rank and accuse the most probable traitor in $\mathcal{T}^{\diamond}$, we record for each user $j \in \mathcal{U}^{(t)}$ the subset leading to that user's highest score:

$$\mathcal{T}_j^{\diamond} = \arg\max_{\mathcal{T}} \{s_{\mathcal{T}} | j \in \mathcal{T}\}. \tag{15}$$

We can count how often each user j appears in the recorded subsets $\{\mathcal{T}_j^{\diamond}\}_{j\in\mathcal{U}^{(t)}}$ and denote this value a_j. Finally, for a given $\mathcal{T}$, the users $j_k \in \mathcal{T}$ can be arranged s.t. $a_{j_1} \geq a_{j_2} \geq \dots \geq a_{j_t}$ to establish a ranking of users per subset.[3]

[3] This detail is omitted in Alg. 1 but necessary for procedure `top()`.

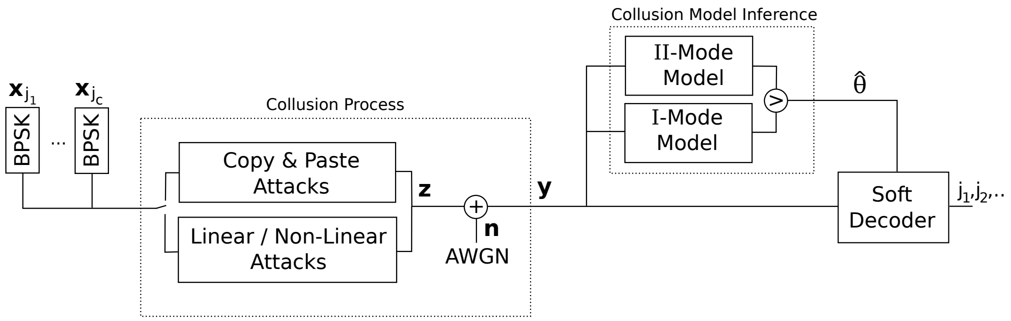


Fig. 3. Attack channel and collusion model inference.

To accuse a user $j \in \mathcal{T}^\diamond$, we check if the single score $s(\mathbf{x}_j, \mathbf{y}, \mathbf{p}, \mathcal{U}_{\mathsf{SI}}\cup\{\mathcal{T}^\diamond\backslash j\}) > \tau'$ with τ' s.t. $\mathbb{P}(s(\mathbf{x}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p}, \mathcal{U}_{\mathsf{SI}}\cup\{\mathcal{T}^\diamond \setminus j\}) > \tau') = n^{-1}P_{\mathsf{fp}}$. This method is suggested in [9, Sec. 5.3].

D. Inference of the collusion process

The MLE is used to infer about the collusion process:

$$\hat{\boldsymbol{\theta}}_{c_{\max}} = \arg \max_{\boldsymbol{\theta}\in\Theta_{c_{\max}}} \log \mathbb{P}(\mathbf{y}|\mathbf{p}, \mathcal{U}_{\mathsf{SI}}, \boldsymbol{\theta}). \tag{16}$$

Whenever a user is deemed guilty, it is added to side-information and we re-run the parameter estimation to refine the collusion inference.

V. Soft Decoding under AWGN attack

The marking assumption is an unrealistic restriction for traitor tracing with multimedia content as the colluders are not limited to the copy-and-paste strategy for each symbol. They can merge the samples of their content versions (audio samples, pixels, DCT coefficients, etc.) in addition to traditional attempts to compromise the watermark. This may result in erroneously decoded symbols or erasures from the watermarking layer. Relaxing the marking assumption leads to several approaches such as the combined digit model [19] [20, Sec. 4] and soft-decision decoding schemes [21], [22]. This section extends the capability of our joint decoder to this latter case, replacing the probability transition $2 \times (c+1)$ matrix $[\mathbb{P}(Y|\Phi)]$ (see Sec. II-B) by $c+1$ probability density functions $\{\theta_c(y|\varphi)\}_{\varphi=0}^{c}$.

It is challenging if not impossible to exhibit a model encompassing all the merging attacks while being relevant for a majority of watermarking techniques. Our approach as sketched in Fig. 3 is pragmatic. The sequence $\mathbf{y}' \in \mathbb{R}^m$ is extracted from the pirated copy, with modulation $y'(i) = 2y(i) - 1$ if the signal is perfectly watermarked with binary symbol $y(i)$. To reflect the merging attack, the colluders forge values $z(i) \in [-1, 1]$ and add noise: $y'(i) = z(i) + n(i)$ with $n(i) \sim \mathcal{N}(0, \sigma_{\mathsf{n}}^2)$. This would be the case, for instance, for a spread spectrum watermarking where a symbol is embedded per block of content with an antipodal modulation of a secret carrier [21], [23].

The colluders have two strategies to agree on $\mathbf{z}$. In a first strategy, they collude according to the marking assumption (*i.e.* they copy-and-paste one of their samples) and add noise: $\mathbf{z} \in \{-1, 1\}^m$ and the probability that

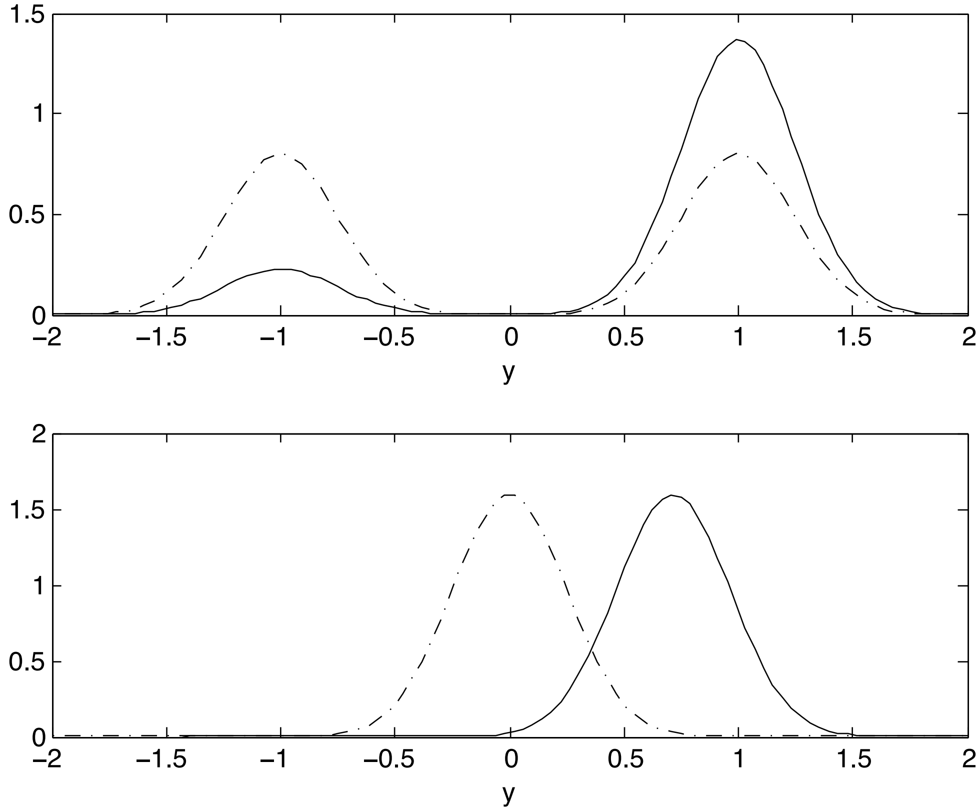


Fig. 4. Examples of pdf $\theta_7(y'|6)$ for the two models with $\sigma_{\mathsf{n}}^2 = 0.25$: [top] two modes (II) with (solid) the interleaving attack ($\theta(\varphi) = \varphi/c$) and (dashed) the coin-flip attack ($\theta(\varphi) = 1/2$ for $0 < \varphi < c$) ; (bottom) one mode (I) with (solid) averaging attack ($\mu(\varphi) = 2c^{-1}\varphi - 1$) and (dashed) set to 0 attack ($\mu(\varphi) = 0$ for $0 < \varphi < c$).

$z = 1$ is given by the components of $\boldsymbol{\theta}_c$.

$$\theta_c^{(II)}(y'|\varphi) = \left(\theta_c(\varphi)e^{-\frac{(y'-1)^2}{2\sigma_{\mathsf{n}}^2}} + (1-\theta_c(\varphi))e^{\frac{(y'+1)^2}{2\sigma_{\mathsf{n}}^2}}\right)/\sqrt{2\pi\sigma_{\mathsf{n}}^2} \tag{17}$$

Except for $\varphi \in \{0, c\}$, the pdfs have a priori two modes (hence the superscript II). This model is parameterized by $(\boldsymbol{\theta}, \sigma_{\mathsf{n}}^2)$.

In a second strategy, the colluders select $z(i) = \mu(\varphi(i)) \in [-1, 1]$:

$$\theta_c^{(I)}(y'|\varphi) = e^{-\frac{(y'-\mu(\varphi))^2}{2\sigma_{\mathsf{n}}^2}}/\sqrt{2\pi\sigma_{\mathsf{n}}^2}. \tag{18}$$

An equivalent of the marking assumption would impose that $\mu(0) = -1$ and $\mu(c) = 1$. The pdfs have a unique mode (hence the superscript I). This model is parameterized by $(\boldsymbol{\mu}, \sigma_{\mathsf{n}}^2)$. Fig. 4 gives some examples of such pdfs.

A simple approach, termed *hard* decision decoding in the sequel, consists in first thresholding $\mathbf{y}'$ (to quantize $y'(i)$ into 0 if $y'(i) < 0$ and 1 otherwise), and then employ the collusion process inference of Sec. IV-D on the hard outputs. Our *soft* decision decoding method resorts to the noise-aware models (17) and (18) and sets

$$\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} = \underset{\boldsymbol{\theta}\in\{\hat{\boldsymbol{\theta}}^{(II)}_{c_{\mathsf{max}}},\hat{\boldsymbol{\theta}}^{(I)}_{c_{\mathsf{max}}}\}}{\arg\max} \mathbb{P}(\mathbf{y}|\mathbf{p},\mathcal{U}_{\mathsf{SI}},\boldsymbol{\theta}). \tag{19}$$

Notice that models I and II share the same number of parameters, therefore, there is no risk of over-fitting.

VI. Experimental Results

We implemented the Tardos decoders in C++[4]. Single and joint score computation is implemented efficiently using pre-computed lookup tables, cf. (12) and (13), and aggregation techniques described in [17]. For a code length of $m = 1024$ more than 10^6 single and about 10^5 joint scores, respectively, can be computed per second on single core of a regular Intel Core2 2.6 GHz CPU. To control the runtime, the joint decoders are confined to 5-subset decoding ($t_{\mathsf{max}} = 5$) and $p^{(t)} \approx 4.5 \cdot 10^6$ computed subsets per joint decoding stage. An iterative decoding experiment can be executed on a PC within a couple of minutes, given enough memory, see [18] for details. To experimentally verify the false-positive rate controlled by rare-event analysis, up to $3\cdot 10^4$ tests per parameter setting have been performed on a cluster of PCs.

First, we first compare the performance of the proposed decoders under marking assumption. Finally, we lift this unrealistic restriction and turn to a more practical assessment using soft-decision decoding.

Unless explicitly noted, the terms *single* and *joint* decoder refer to the decoders conditioned on the inference of the collusion process $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$, cf. (8) and (12). Further, we consider the MAP decoders assuming knowledge of $\boldsymbol{\theta}_c$ and the compound channel decoder, cf. (7), tuned on the worst-case attack $\boldsymbol{\theta}^{\star}_{k,f_T}$, $\forall k \in [2, \ldots, c_{\mathsf{max}}]$. As a baseline for a performance comparison, we always include symmetric Tardos score computation [8] with a threshold controlled by rare-event analysis (see Sec. III-C).

A. Decoding performance under marking assumption

1) Detect-one scenario: Here the aim is to catch at most one colluder – this is the tracing scenario most commonly considered in the literature. We compare our *single* and *joint* decoder performance against the results provided by Nuida *et al.* [24] (which are the best as far as we know) and, as a second reference, the symmetric Tardos decoder.

The experimental setup considers $n = 10^6$ users and $c \in \{2, 3, 4, 6, 8\}$ colluders performing *worst-case* attack [14] against a single decoder. In Fig. 5, we plot the empirical probability of error $P_{\mathsf{e}} = P_{\mathsf{fp}} + P_{\mathsf{fn}}$ obtained by running 10^4 experiments for each setting versus the code length m. The false-positive error is controlled by thresholding based on rare-event simulation, $P_{\mathsf{fp}} = 10^{-3}$, which is confirmed experimentally. Evidently, for a given probability of error, the *joint* decoder succeeds in reducing the required code length over the *single* decoder, especially for larger collusions.

Table I compares the code length to obtain an error rate of $P_{\mathsf{e}} = 10^{-3}$ for our proposed Tardos decoders and the symmetric Tardos decoder with the results reported by Nuida *et al.* [24] under marking assumption. Except for

[4] Source code is available at http://www.irisa.fr/texmex/people/furon/src.html.

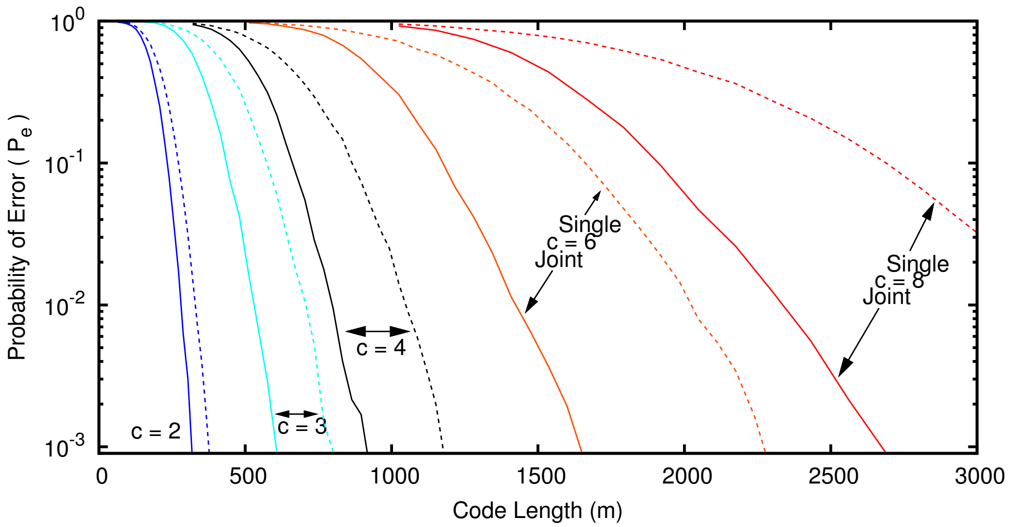


Fig. 5. Code length vs. P_{e} for $n = 10^6$ users and different number of colluders performing *worst-case* attack against a single decoder; $c_{\mathsf{max}} = 8$.

TABLE I

Code length comparison for the detect-one scenario: $n = 10^6$, Worst-Case attack against a single decoder, $P_{\mathsf{e}} = 10^{-3}$.

Colluders	Nuida *et al.*	Symm.	Proposed ($c_{\mathsf{max}} = 8$)	
(c)	[24]	Tardos	Single	Joint
2	253	~ 416	~ 368	~ 304
3	877	~ 864	~ 776	~ 584
4	1454	~ 1472	~ 1152	~ 904
6	3640	~ 2944	~ 2304	~ 1616
8	6815	~ 5248	~ 3712	~ 2688

$c = 2$, the proposed decoders can substantially reduce the required code length and the *joint* decoder improves the results of the *single* decoder. Note that Nuida's results give analytic code length assuming a particular number of colluders for constructing the code while our results are experimental estimates based on worst-case attack against a single decoder and without knowing c (subject to $c \leq c_{\mathsf{max}} = 8$). Results with c known are provided in [18] and show a slightly better performance: the required code length of the *joint* decoder is then slightly shorter than Nuida's code in case $c = 2$.

2) Detect-many scenario: We now consider the more realistic case where the code length m is fixed and the false-negative error rate is only a minor concern[5] while the false-positive probability is critical to avoid an accusation of an innocent. The aim is to identify as many colluders as possible.

[5] A tracing schemes rightly accusing a colluder half of the time might be enough to dissuade dishonest users.

Figures 6(a)–6(d) show the average number of identified colluders by different decoding approaches. The experimental setup considers $n = 10^6$ users, code length $m = 2048$, and several collusion attacks (*worst-case* attacks, i.e. minimizing the achievable rate of a single or joint decoder, *interleaving* and *majority* which is a rather mild attack) carried out by two to eight colluders. The global probability of a false positive error is fixed to $P_{\mathsf{fp}} = 10^{-3}$.

As expected, the MAP single decoder knowing $\boldsymbol{\theta}_c$ provides the best decoding performance amongst the single decoders, yet is unobtainable in practice. The symmetric Tardos decoder performs poorly but evenly against all attacks; the single decoder based on the compound channel (7) improves the results only slightly.

The *joint* decoders consistently achieve to identify most colluders – with a dramatic margin in case the traitors choose the worst-case attack against a single decoder. This attack bothers the very first step of our decoder, but as soon as some side information is available or a joint decoder is used, this is no longer the worst case attack. Finding the worst case attack against our iterative decoder is indeed difficult. A good guess is the interleaving attack which is asymptotically the worst case against the joint decoder [1]. The experiments show that it reduces the performance of the *joint* decoders substantially for large c.

The decoder based on the inference $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$ and the true MAP are different when c is lower than c_{max}. However, this is not a big deal in practice for a fixed m: for small c, the code is long enough to face the collusion even if the score is less discriminative than the ideal MAP; for big c the score of our decoder gets closer to the ideal MAP.

B. Decoding performance of the soft decoder

We assess the performance of the soft decision decoders proposed in Sec. V in two tracing scenarios: (i) Kuribayashi considers in [21] $n = 10^4$ users and code length $m = 10^4$, (ii) a large-scale setup with $33\,554\,432$ users and $m = 7\,440$ where Jourdas and Moulin [23] provide results for their high-rate random-like fingerprinting code under averaging and interleaving attack.

In Fig. 7, we compare the average number of identified colluders for the *single* and *joint* decoder using different estimates of the collusion process: *hard* relates to decoders using hard thresholding and $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$ while *soft* identifies the noise-aware decoders relying on $\hat{\boldsymbol{\theta}}^{(I)}_{c_{\mathsf{max}}}$ or $\hat{\boldsymbol{\theta}}^{(II)}_{c_{\mathsf{max}}}$ chosen adaptively based on the likelihood of the two models. All plots also show the results for the (hard-thresholding) symmetric Tardos decoder. The false-positive rate is set to 10^{-4}. Extensive experiments ($3 \cdot 10^4$ test runs) have been carried out to validate the accusation threshold obtained by rare-event simulation. As expected, soft decoding offers substantial gains in decoding performance. The margin between the *single* and *joint* decoders depends on the collusion strategy. Dramatic improvements can be seen when the collusion chooses the *worst-case* attack against a single decoder, cf. Fig. 7(a). On the other hand, the gain is negligible when averaging is performed.

Note that the attacks in (a)–(c) pertain to the pick-and paste attacks while Fig. 7(d) shows the linear *averaging* attack.

Comparison with the results provided in [21] for the *majority* attack is difficult: (i) they were obtained for Nuida's discrete code construction [24] tuned on $c = 7$ colluders, and (ii) the false-positive rate of [21] does not seem to be under control for the symmetric Tardos code. We suggest to use the *hard* symmetric Tardos decoder [8] as a baseline for performance comparison. By replacing the accusation thresholds proposed in [21] with a rare-event simulation, we are able to fix the false-alarm rate in case of the symmetric Tardos code. Furthermore, the decoding results given in [21] for the discrete variant of the fingerprinting code (*i.e.* Nuida's construction) could be significantly improved by rare-event simulation based thresholding. Contrary to the claim of [21], *soft* decision decoding always provides a performance benefit over the *hard* decoders.

In Fig. 8 we illustrate the decoding performance when dealing with a large user base. We consider *averaging* and *interleaving* attacks by $c = 2, \ldots, 12$ and $c = 2, \ldots, 8$ colluders ($c_{\mathsf{max}} = 12$ and $c_{\mathsf{max}} = 8$, respectively) followed by AWGN with variance $\sigma_{\mathsf{n}}^2 = 1$. The global false-positive rate is set up to 10^{-3}. The benefit of the *soft* decoding approach in clearly evident. Joint decoding provides only a very limited increase in the number of identified colluders. For comparison, Jourdas & Moulin indicate an error rate of $P_{\mathsf{e}} = 0.0074$ for $c = 10$ colluders in the first, and $P_{\mathsf{e}} = 0.004$ for $c = 5$ colluders in the second setting for a detect-one scenario [23].

In [25], $P_{\mathsf{fp}} = 0.0016$ and $P_{\mathsf{fn}} = 0.044$ are given for the first experiment (Fig. 8(a)) by introducing a threshold to control the false-positive rate. Our *soft joint* decoder achieves a $P_{\mathsf{fn}} = 0.046$ for $P_{\mathsf{fp}} = 10^{-3}$ (for $c = 10$ colluders), catching 2.6 traitors on average.

In the second experiment (see Fig. 8(b)), our *joint* decoder compares more favorably: with the given code length, all $c = 5$ colluders can be identified and for a collusion size $c = 8$, 4.5 traitors are accused without observing any decoding failure in $3 \cdot 10^3$ tests.

C. Runtime Analysis

Single decoding can be efficiently implemented to compute more than one million scores for a code of length $m = 1024$ per second. Its complexity is in $O(n \cdot m)$. Selecting the $p^{(t)}$ most likely guilty users can be efficiently done with the max-heap algorithm. Yet, it consumes a substantial parts of the runtime for small m. The runtime contribution of the joint decoding stage clearly depends on the size of pruned list of suspects, $O(m \cdot p^{(t)})$ and is independent of the subset size t thanks to the *revolving door* enumeration method of the subsets[6]. Restricting $p^{(t)}$ and t_{max} keeps the joint decoding approach computationally tractable. Better decoding performance can be obtained using higher values at the cost of a substantial increase in runtime. Experiments have shown that even the moderate settings ($p^{(t)} \approx 4.5 \cdot 10^6$ and $t_{\mathsf{max}} = 5$) achieve a considerable gain of the joint over the single decoder for several collusion channels.

Thresholding accounts for more than half of the runtime in the experimental setups investigated in this work. However, this is not a serious issue for applications with a large user base or when $p^{(t)}$ becomes large. Thresholding depends on the subset size t because a large number of random codeword combinations must be generated and because we seek lower probability level in $O(P_{\mathsf{fp}}/n^t)$. Therefore, the complexity is in $O(m \cdot t^2 \cdot \log(n))$ according to [16]. There are no more than c_{max} such iteration with $t \leq c_{\mathsf{max}}$, so that the global complexity of our decoder stays in $O(m \log(n))$.

[6] In each step $\boldsymbol{\varphi}$ is updated by replacing one user's codeword. See [18] for details.

More details about the runtime are given in [18]. Note that results have been obtained with a single CPU core although a parallel implementation can be easily achieved.

VII. Conclusion

Decoding probabilistic fingerprinting codes in practice means to trace guilty persons over a large set of users while having no information about the size nor the strategy of the collusion. This must be done reliably by guaranteeing a controlled probability of false alarm.

Our decoder implements provably good concepts of information theory (joint decoding, side information, linear decoder for compound channels) and statistics (estimation of extreme quantile of a rare event). Its extension to soft output decoding is straightforward as its does not change its architecture.

Since the proposed iterative method is neither just a single decoder nor completely a joint decoder (it only considers subsets over a short list of suspects), it is rather difficult to find the best distribution for code construction and its worst case attack. Experiments show that the interleaving attack is indeed more dangerous than the worst-case attack against a single decoder.

Appendix

We prove that $\mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c) = \{\tilde{\boldsymbol{\theta}}_k | k \leq c_{\mathsf{max}}, \mathbb{P}(y|p, \tilde{\boldsymbol{\theta}}_k) = \mathbb{P}(y|p, \boldsymbol{\theta}_c), \forall (y,p) \in \{0,1\} \times [0,1]\}$ is one sided. The collusion channels of this set share the property that $\mathbb{P}(Y=1|p, \tilde{\boldsymbol{\theta}}_k) = q(p) \geq 0, \forall p \in [0,1]$. From [14, Eq. (20)]:

$$\mathbb{P}(Y=1|X=1,p,\tilde{\boldsymbol{\theta}}_k) = q(p) + k^{-1}(1-p)q'(p) \tag{20}$$

$$\mathbb{P}(Y=1|X=0,p,\tilde{\boldsymbol{\theta}}_k) = q(p) - k^{-1}pq'(p) \tag{21}$$

Take $(\tilde{\boldsymbol{\theta}}_{k_A}, \tilde{\boldsymbol{\theta}}_{k_B}) \in \mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c)^2$ s.t. $k_A < k_B$. We first show that $R(f_T, \tilde{\boldsymbol{\theta}}_{k_A}) > R(f_T, \tilde{\boldsymbol{\theta}}_{k_B})$ so that the minimizer of $R(f_T, \boldsymbol{\theta})$ over $\mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c)$ is indeed $\tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$. Denote by (μ_1, μ_2) the following conditional probability distributions:

$$\mu_1(y,x|p) = \mathbb{P}(Y=y|p) = q(p)^y(1-q(p))^{(1-y)} \tag{22}$$

$$\mu_2(y,x|p) = \mathbb{P}(Y=y|X=x,p,\tilde{\boldsymbol{\theta}}_{k_A}). \tag{23}$$

Then, $\mathbb{P}(Y|X,p,\tilde{\boldsymbol{\theta}}_{k_B}) = (1-\lambda)\mu_1(Y,X|p) + \lambda\mu_2(Y,X|p)$, $\forall p \in [0,1]$, with $\lambda = k_A/k_B < 1$. The mutual information is a convex function of $\mathbb{P}(Y|X,p)$ for fixed $\mathbb{P}(X|p)$ so that, once integrated over $f_T(p)$, we have

$$R(f_T, \tilde{\boldsymbol{\theta}}_{k_B}) \leq (1-\lambda)\cdot 0 + \lambda \cdot R(f_T, \tilde{\boldsymbol{\theta}}_{k_A}) < R(f_T, \tilde{\boldsymbol{\theta}}_{k_A}). \tag{24}$$

We now prove that (6) holds $\forall \boldsymbol{\theta} \in \mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c)$. This is equivalent to

$$R(f_T, \tilde{\boldsymbol{\theta}}_k) - D(\mathbb{P}(Y,X|\tilde{\boldsymbol{\theta}}_k)||\mathbb{P}(Y,X|\tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}})) - R(f_T, \tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}) \geq 0, \tag{25}$$

where the LHS is of the form $\mathbb{E}_{P\sim f_T}[g(P)]$. After developing the expressions, we find that:

$$\begin{aligned} g(p) \quad = \quad & (k^{-1} - c_{\mathsf{max}}^{-1})p(1-p) \cdot \\ & \left(q'(p) \log \left(1 + \frac{1-p}{c_{\mathsf{max}}} \frac{q'(p)}{q(p)}\right) + \right. \\ & q'(p) \log \left(1 + \frac{p}{c_{\mathsf{max}}} \frac{q'(p)}{1-q(p)}\right) - \\ & q'(p) \log \left(1 - \frac{1-p}{c_{\mathsf{max}}} \frac{q'(p)}{1-q(p)}\right) - \\ & \left. q'(p) \log \left(1 - \frac{p}{c_{\mathsf{max}}} \frac{q'(p)}{q(p)}\right) \right) \end{aligned} \tag{26}$$

The four terms inside parenthesis are not negative because, with $\gamma > 0$, $x\log(1+\gamma x) \geq 0$ for $x > -\gamma^{-1}$. Since $k \leq c_{\mathsf{max}}$, we obtain $g(p) \geq 0$, whence (6).

References

[1] Y.-W. Huang and P. Moulin, "On the saddle-point solution and the large-coalition behavior of fingerprinting games," *IEEE Transactions on Information Forensics and Security*, Apr. 2011, submitted, arXiv:1011.1261v2.

[2] E. Amiri and G. Tardos, "High rate fingerprinting codes and the fingerprinting capacity," in *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '09*. New York, NY, USA: SIAM, Jan. 2009, pp. 336–345.

[3] N. P. Anthapadmanabhan, A. Barg, and I. Dumer, "On the fingerprinting capacity under the marking assumption," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2678–2689, Jun. 2008.

[4] E. Abbe and L. Zheng, "Linear universal decoding for compound channels," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 5999–6013, Dec. 2010.

[5] G. Tardos, "Optimal probabilistic fingerprint codes," *Journal of the ACM*, vol. 55, no. 2, pp. 1–24, May 2008.

[6] A. Barg, G. R. Blakley, and G. Kabatiansky, "Digital fingerprinting codes: Problem statements, constructions, identification of traitors," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 852–865, Apr. 2003.

[7] M. Fernandez and M. Soriano, "Identification of traitors in algebraic-geometric traceability codes," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3073–3077, Oct. 2004.

[8] B. Skoric, S. Katzenbeisser, and M. Celik, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," *Designs, Codes and Cryptography*, vol. 46, no. 2, pp. 137–166, Feb. 2008.

[9] P. Moulin, "Universal fingerprinting: Capacity and random-coding exponents," May 2011, arXiv:0801.3837v3.

[10] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Transaction on Information Theory*, vol. 44, no. 5, pp. 1897–1905, September 1998.

[11] T. Furon and L. Pérez-Freire, "EM decoding of Tardos traitor tracing codes," in *Proceedings of the ACM Multimedia Security Workshop*, Princeton, NJ, USA, Sep. 2009, pp. 99–106.

[12] T. Furon, A. Guyader, and F. Cérou, "On the design and optimisation of Tardos probabilistic fingerprinting codes," in *Proceedings of the 10th Information Hiding Workshop*, ser. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, May 2008, pp. 341–356.

[13] A. Somekh-Baruch and N. Merhav, "On the capacity game of private fingerprinting systems under collusion attacks," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 884–899, 2005.

[14] T. Furon and L. Pérez-Freire, "Worst case attacks against binary probabilistic traitor tracing codes," in *Proceedings of the First IEEE International Workshop on Information Forensics and Security*. London, UK: WIFS'09, Dec. 2009, pp. 46–50.

[15] Y.-W. Huang and P. Moulin, "Capacity-achieving fingerprint decoding," in *Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS '09*, London, UK, Dec. 2009, pp. 51–55.

[16] A. Guyader, N. Hengartner, and E. Matzner-Lober, "Simulation and estimation of extreme quantiles and extreme probabilities," *Applied Mathematics & Optimization*, vol. 64, no. 2, pp. 171–196, 2011, http://www.sites.univ-rennes2.fr/laboratoire-statistique/AGUYADER/doc/ghm.pdf.

[17] P. Meerwald and T. Furon, “Iterative single Tardos decoder with controlled probability of false positive,” in *Proceedings of the IEEE International Conference on Multimedia & Expo, ICME ’11*, Barcelona, Spain, Jul. 2011.

[18] ——, “Towards joint Tardos decoding: The ’Don Quixote’ algorithm,” in *Proceedings of the Information Hiding Conference, IH ’ 11*, ser. Lecture Notes in Computer Science, vol. 6958. Prague, Czech Republic: Springer, May 2011, pp. 28–42.

[19] B. Skoric, S. Katzenbeisser, H. Schaathun, and M. Celik, “Tardos fingerprinting codes in the combined digit model,” in *Proceedings of the First IEEE International Workshop on Information Forensics and Security*. London, UK: WIFS’09, Dec. 2009, pp. 41–45.

[20] L. Pérez-Freire and T. Furon, “Blind decoder for binary probabilistic traitor tracing codes,” in *Proceedings of the First IEEE International Workshop on Information Forensics and Security*. London, UK: WIFS’09, Dec. 2009, pp. 56–60.

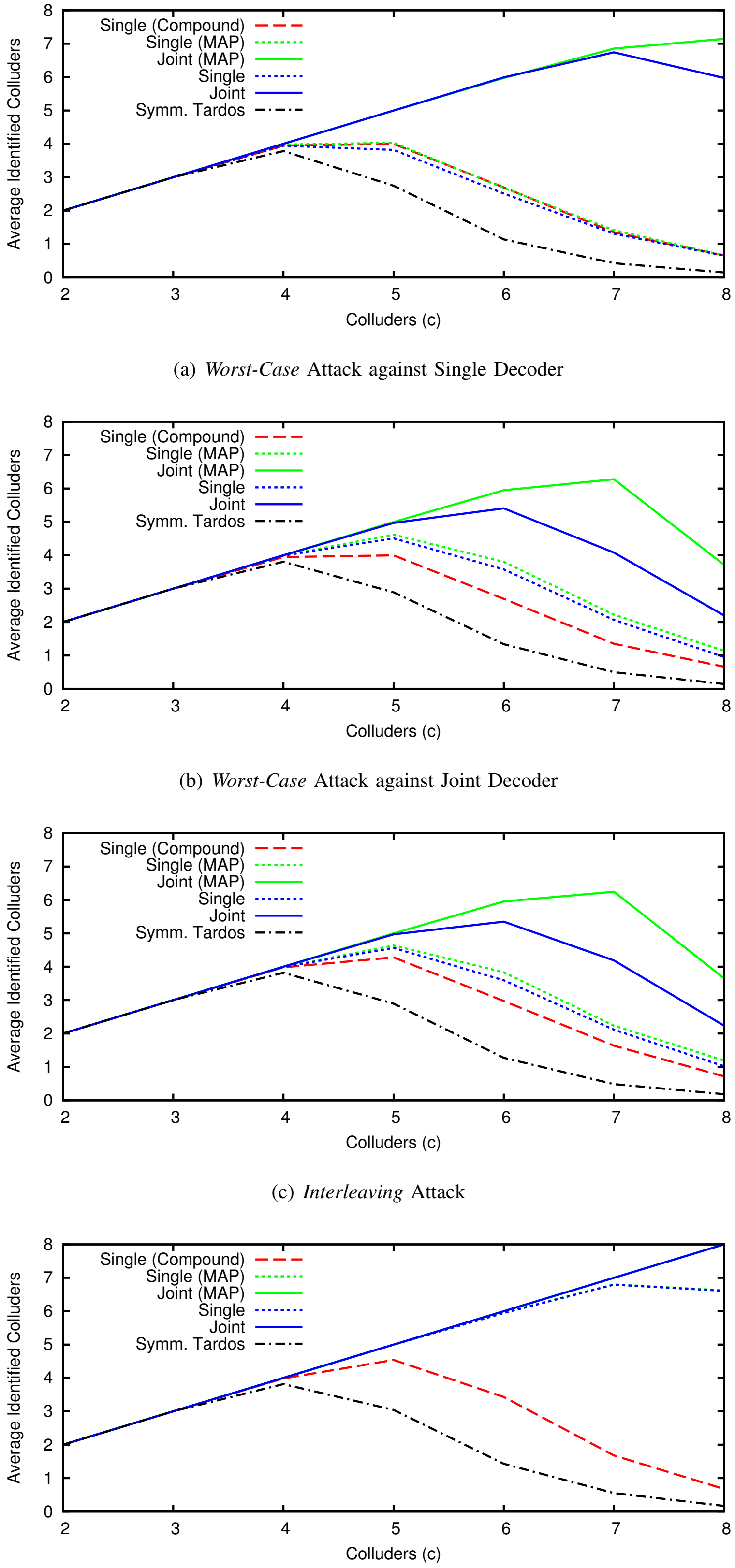
[21] M. Kuribayashi, “Experimental assessment of probabilistic fingerprinting codes over AWGN channel,” in *Proceedings of the 5th International Workshop on Security, IWSEC ’10*, ser. Lecture Notes in Computer Science, vol. 6432. Kobe, Japan: Springer, Nov. 2010, pp. 117–132.

[22] H. G. Schaathun, “On error-correcting fingerprinting codes for use with watermarking,” *Multimedia Systems*, vol. 13, no. 5, pp. 331–344, 2008.

[23] J.-F. Jourdas and P. Moulin, “High-rate random-like spherical fingerprinting codes with linear decoding complexity,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 768–780, Dec. 2009.

[24] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, “An improvement of discrete Tardos fingerprinting codes,” *Designs, Codes and Cryptography*, vol. 52, no. 3, pp. 339–362, Mar. 2009, http://eprint.iacr.org/2008/338.

[25] J.-F. Jourdas and P. Moulin, “A high-rate fingerprinting code,” in *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging, Security, Forensics, Steganography and Watermarking of Multimedia Contents X*, San Jose, CA, USA, Jan. 2008.



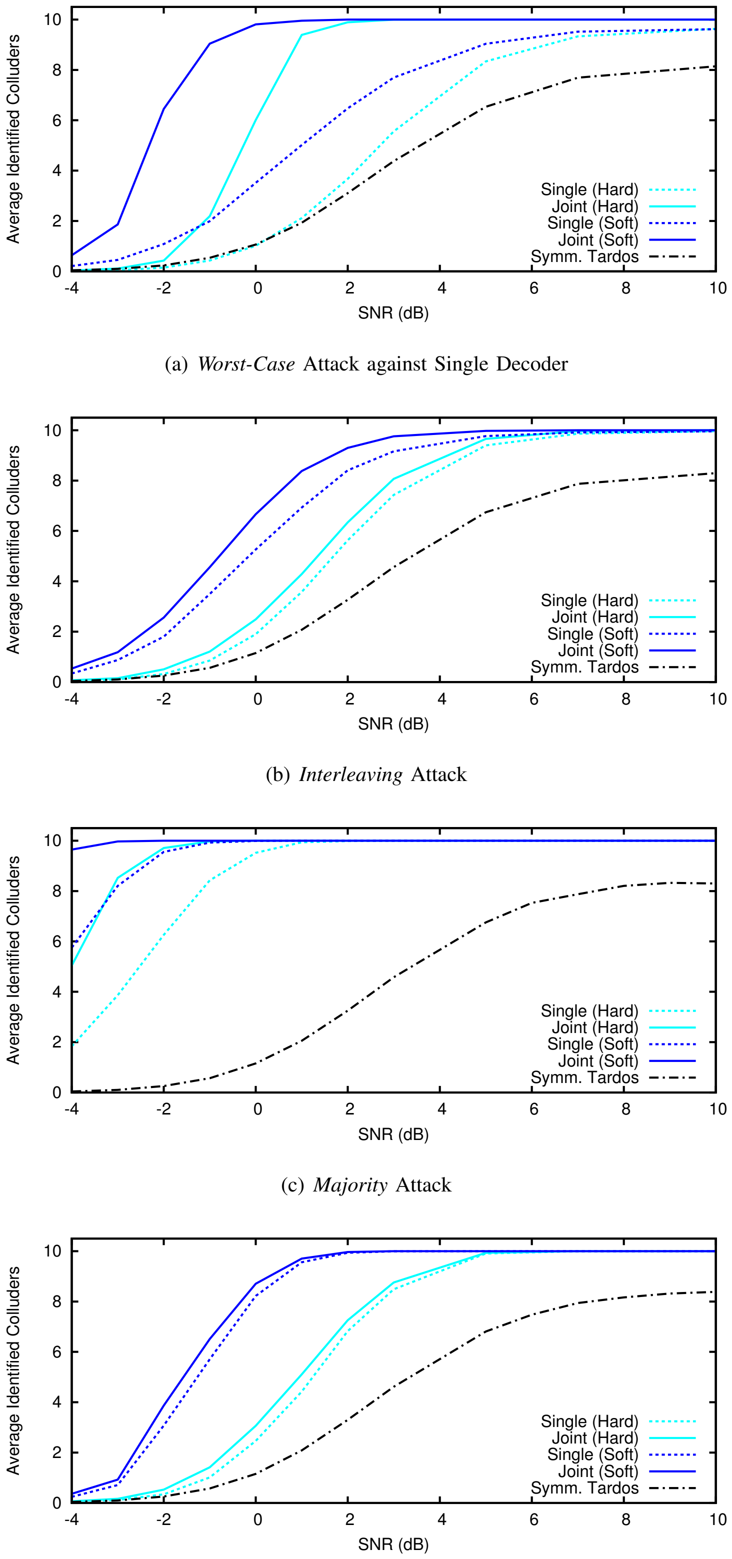
(a) *Worst-Case* Attack against Single Decoder

(b) *Worst-Case* Attack against Joint Decoder

(c) *Interleaving* Attack

(d) *Majority* Attack

Fig. 6. Decoder comparison in the detect-many tracing scenario: $n = 10^6$, $m = 2048$, $P_{\mathsf{fp}} = 10^{-3}$, $c_{\mathsf{max}} = 8$. (Best viewed in color.)



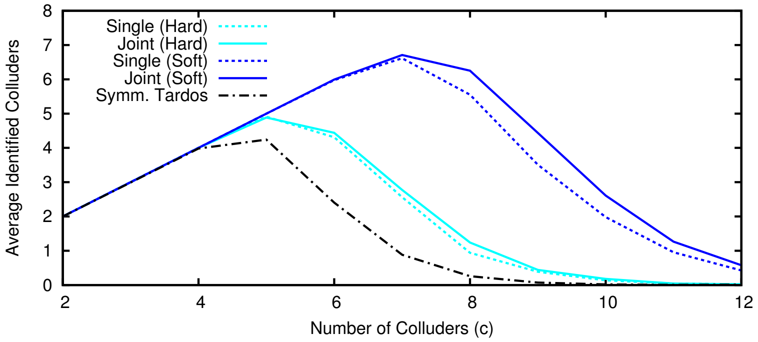
(a) *Worst-Case* Attack against Single Decoder

(b) *Interleaving* Attack

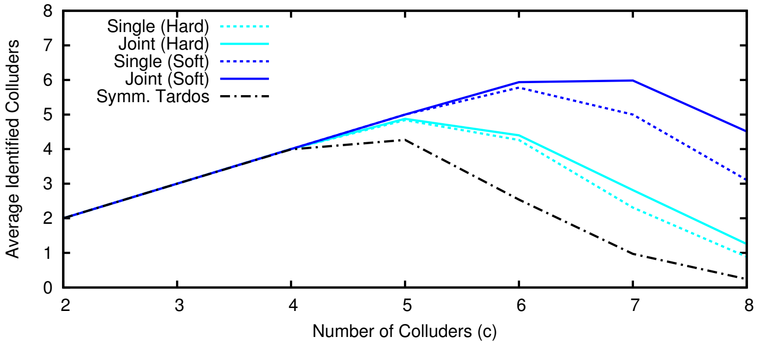
(c) *Majority* Attack

(d) *Averaging* Attack

Fig. 7. Kuribayashi setup: $n = 10^4$, $m = 10^4$, $P_{\mathsf{fp}} = 10^{-4}$, $c = 10$, $c_{\mathsf{max}} = 20$; *worst-case*, *interleaving*, *majority* and *averaging* attack followed by AWGN $(-4, \ldots, 10$ dB SNR).



(a) *Averaging* Attack



(b) *Interleaving* Attack

Fig. 8. Jourdas & Moulin setup: $n = 33\,554\,432$, $m = 7\,440$, $P_{\mathsf{fp}} = 10^{-3}$, *averaging* and *interleaving* attack followed by AWGN (0 dB SNR).